# 实验五 自定义配置文件及JSR303校验的使用

配置文件:系统自动扫描的全程配置文件application.properties 或application.yaml; 额外加载的自定义的配置文件

自定义的配置文件：

properties配置文件：需要使用@PropertiSource加载

xml配置文件：@ImportResource加载

1. 使用Spring Initializr方式新建Spring Boot项目或在原项目上进行下列修改

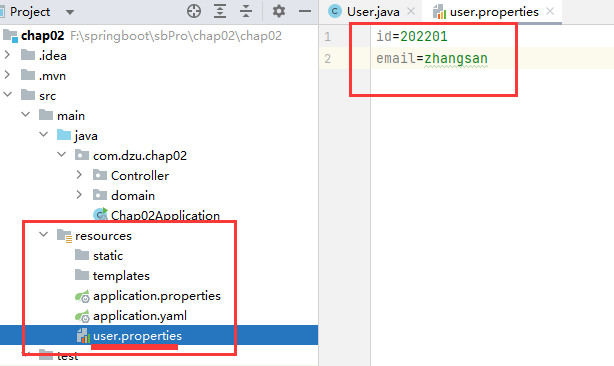
1.新建项目

2.选择Spring Web依赖

3.设置Maven的配置文件和本地仓库

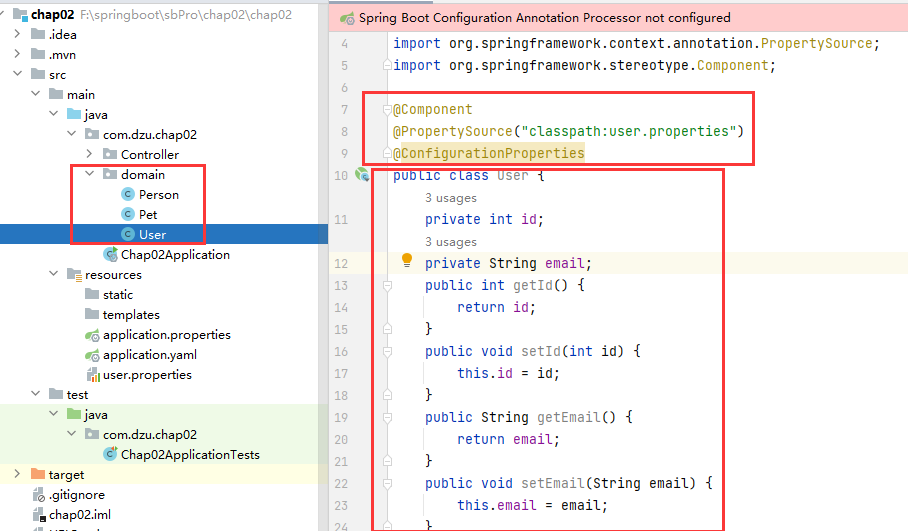
4.在pom.xml中修改Java版本

二、在resourses下新建user.properties配置文件

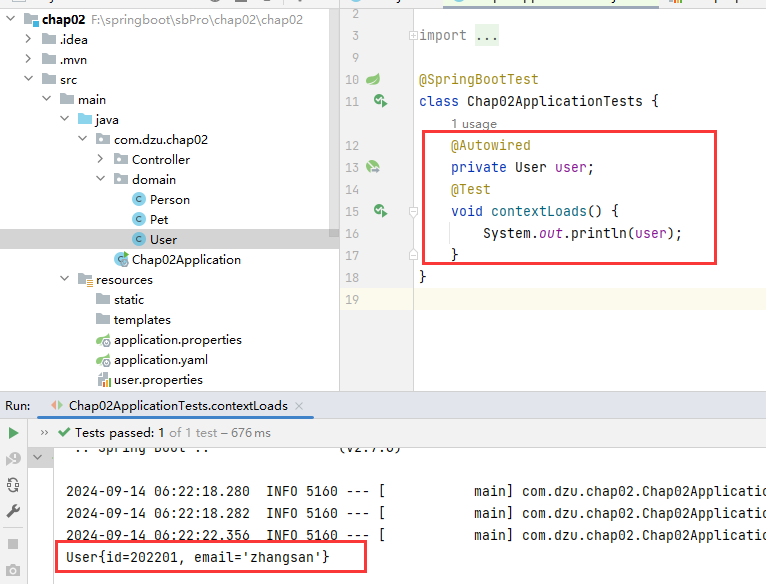


三、创建User实体类

声明id、email属性，并自动生成getter、setter、toString方法。[在类上添加@Component、@PropertySource(“classpath:user.properties)、@ConfiguratinProperties](mailto:在类上添加@Component、@PropertySource()注解

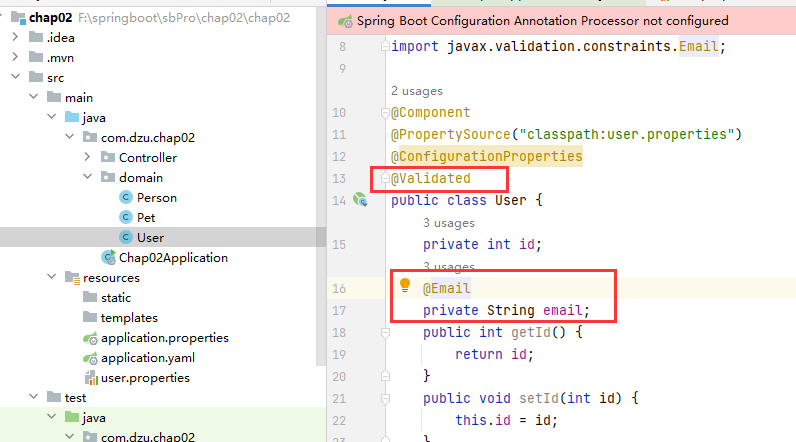


四、新增测试方法或编写测试类，并运行测试类中的测试方法



五、JSR303数据校验

[1.为User类添加@Validate](mailto:1.为User类添加@Validate)注解，并为email属性添加@Email注解

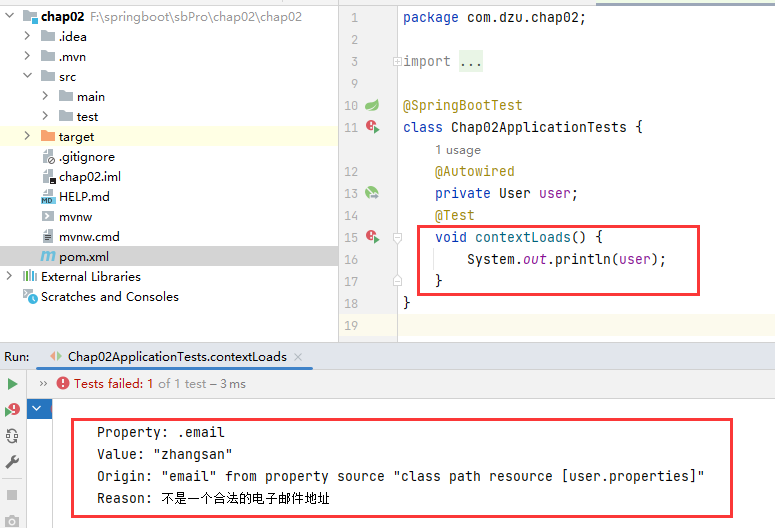


2.在pom.xml中添加validation依赖

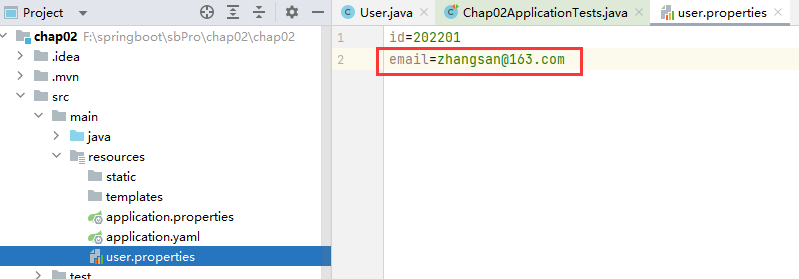
<dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-validation</artifactId>  
</dependency>



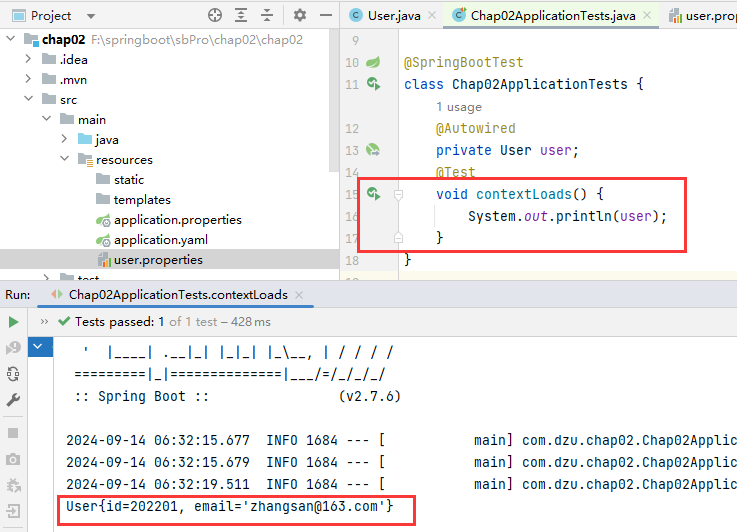
3.运行测试方法，程序出错



4.修改user.properties文件，将email修改为正确的格式

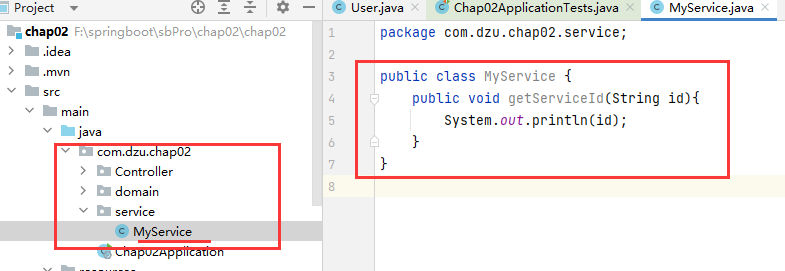


5.重写运行测试方法



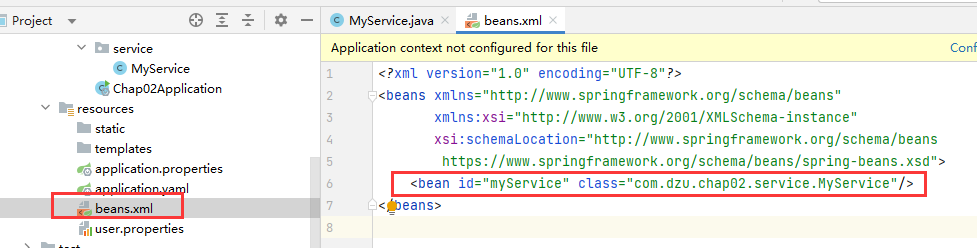
六、使用@ImportResource引入XML配置文件

1.创建MyService类

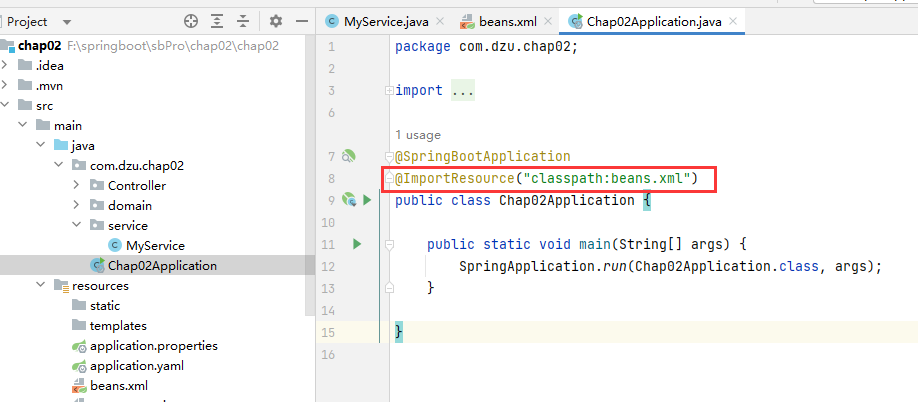


2.创建XML配置文件

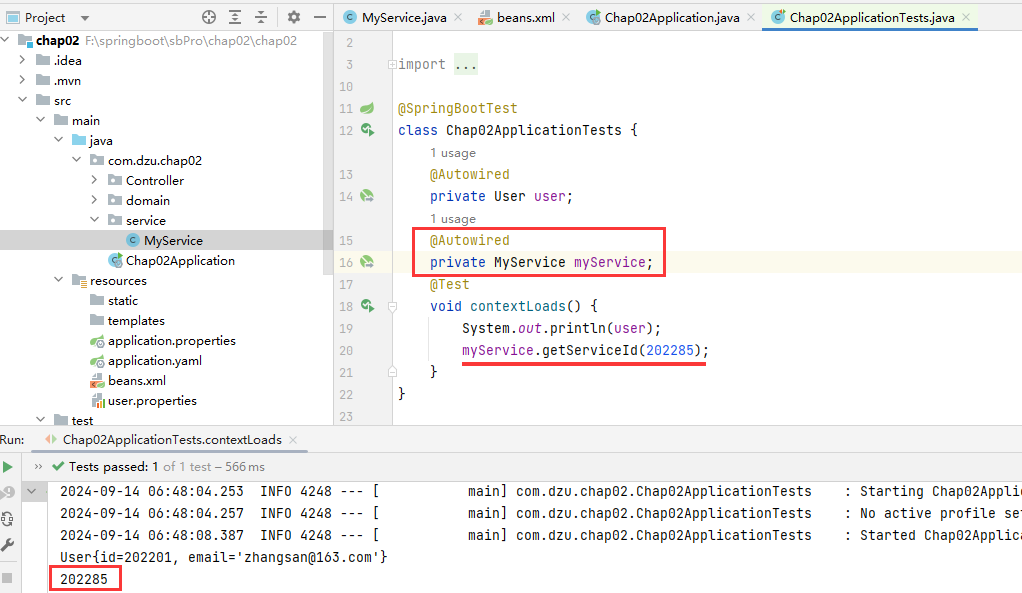
*<?*xml version="1.0" encoding="UTF-8"*?>*<beans xmlns="http://www.springframework.org/schema/beans"  
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="http://www.springframework.org/schema/beans  
 https://www.springframework.org/schema/beans/spring-beans.xsd">  
 <bean id="myService" class="com.dzu.chap02.service.MyService"/>  
</beans>



3.在启动类上添加@ImportResource注解

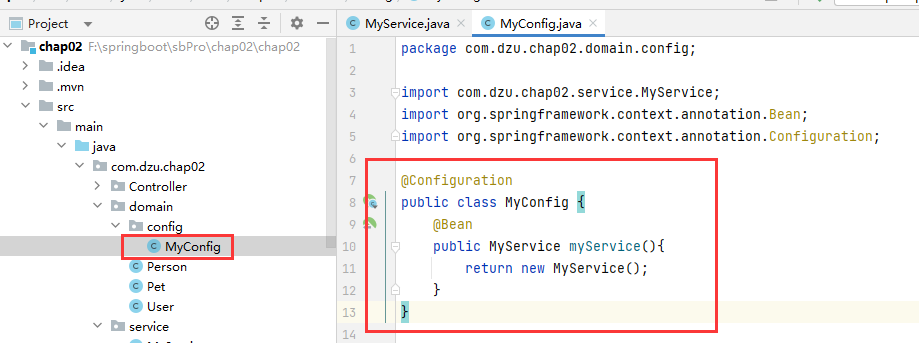


4.新增测试方法并运行



七、自定义配置类

1.创建配置类



[2.注释掉启动类上的@ImportResource](mailto:2.注释掉启动类上的@ImportResource)注解



3.重新运行测试方法

